

Trabajo Práctico 1 — Java

[7507/9502] Algoritmos y Programación III
Curso 2
Primer cuatrimestre de 2022

Integrante	Padrón
Lucas Franciulli	107059
Mario Santiago Adris	106870
Agustín Ezequiel Cáceres	105613
Tomás Benítez Potochek :	100862

Índice

1. Introducción	2
2. Supuestos	2
3. Diagramas de clase	2
4. Detalles de implementación	2
5. Excepciones	3
6. Diagramas de secuencia	3

1. Introducción

El objetivo de este trabajo práctico número 2 es poder desarrollar una aplicación de manera grupal aplicando todos los conceptos vistos en el curso. En este trabajo se usó el lenguaje de tipado estático (Java) con un diseño del modelo orientado a objetos y trabajamos con las técnicas de Test Driven Development e integración continua. Más abajo, se explicarán los supuestos, la implementación y el diseño.

2. Supuestos

- Puede haber más de un vehículo en el mismo casillero.
- Cuando un vehículo pasa por un obstáculo se le agregará a la cantidad de movimientos totales la penalización más uno.

3. Diagramas de clase

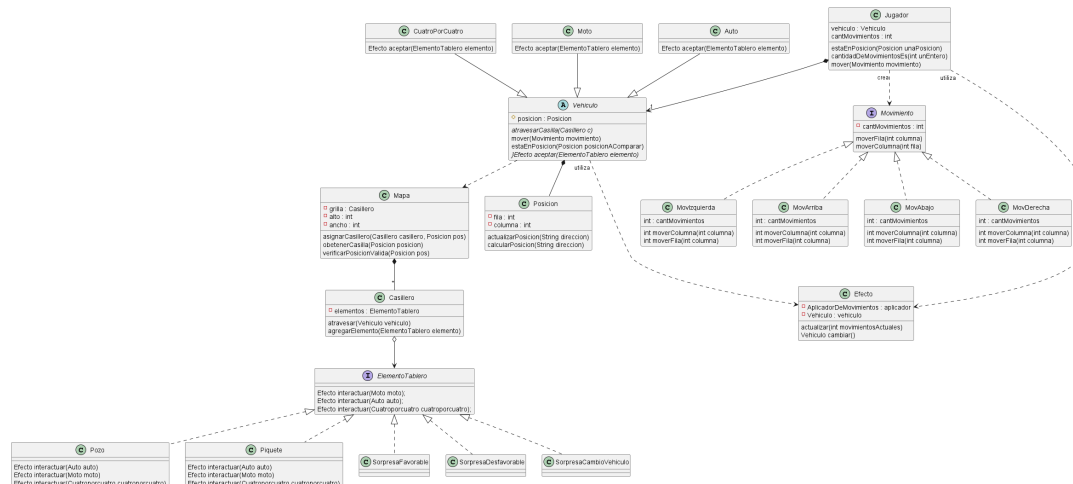


Figura 1: Diagrama del GPS-Challenge.

4. Detalles de implementación

Para el diseño del mapa decidimos hacer uso de un HashMap el cual nos facilitará almacenar las casillas de todo el tablero. Para poder saber dónde guardarlo se hace uso de la función de Hash a la posición del casillero. De esta forma cuando queramos acceder a este casillero simplemente necesitaremos tener la posición. Aunque esto nos trae muchos beneficios también necesitamos saber cuándo el vehículo quiere moverse fuera del tablero para poder negarle esa acción. La forma en la que lo representamos fue asumiendo que sus dimensiones son cuadradas (es decir alto y ancho tienen el mismo valor). Nosotros pensamos el tablero con la posición (1,1) siendo la esquina superior izquierda y si se quiere mover el vehículo para abajo este debe cambiar su posición aumentándole uno a su fila quedando: (2, 1). Entonces a través de un método de mapa simplemente verifica si la posición en la que se quiere mover el vehículo esta dentro de las dimensiones del tablero.

Para la implementación del mapa aplicamos el patrón singleton poniendo el constructo del mismo en privado, y con una variable estática que guarda una referencia a la única instancia del mapa. Para obtener la única instancia del mapa se hará uso del método estático getMapa que en

el caso de que no exista todavía una referencia al mapa este método entonces devolverá una nueva instancia

```
public static Mapa getMapa(){
    if(unMapa == null) {
        unMapa = new Mapa();
    }
    return unMapa;
}

public void asignarCasillero(Casillero casillero, Posicion pos){
    this.grilla.put(pos, casillero);
}
```

5. Excepciones

NoPuedeAtravesarObstaculoError Se lanzará en el caso de que un vehículo no pueda atravesar un obstaculo determinado.

PosicionInvalida Esta excepción se lanzará en caso de que el vehículo quiera moverse en una dirección en la cual no hay una casilla.

6. Diagramas de secuencia

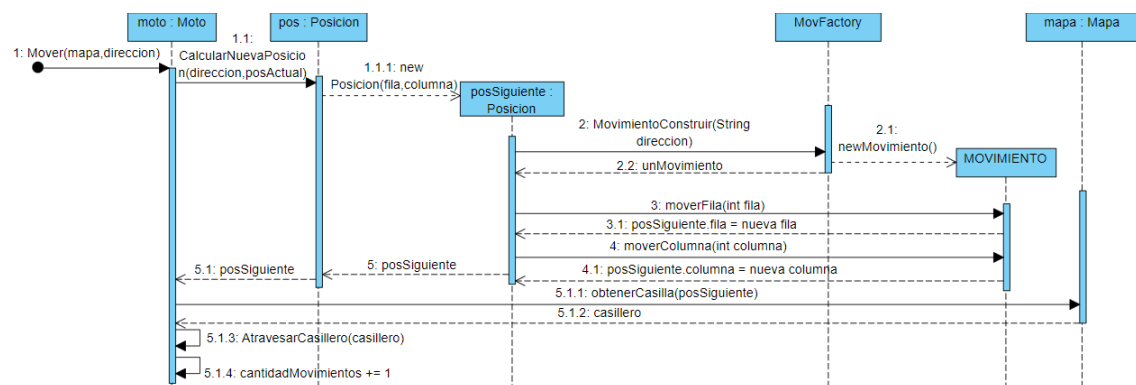


Figura 2: Secuencia de una moto moviéndose a través del mapa